

- Cada ejercicio se califica con Bien, Regular o Mal. La división de los ejercicios en incisos es meramente orientativa. Los ejercicios se califican globalmente.
- El parcial NO es a libro abierto, pero puede tener los apuntes provistos por la cátedra y una hoja A4 con apuntes propios.
- **Importante:** Justifique sus respuestas. Las soluciones a ejercicios de la práctica que se utilicen deben ser incluidas en el examen.
- El parcial se aprueba con los ejercicios 1 y 2 Bien y del 3 y 4 uno Bien o ambos Regular.

**Ejercicio 1** La RECUP-LAST<sup>1L</sup> es una máquina con arquitectura Von Neumann que opera con palabras e instrucciones de 8 bits. Tiene 4 registros de 8 bits de propósito general, una memoria direccionable a palabra y opera con aritmética de complemento a 2. El set de instrucciones es el siguiente:

Instrucción	Formato	opCode	Efecto
ADD Rx, Ry	R	0000	$Rx \leftarrow Rx + Ry$
SUB Rx, Ry	R	0001	$Rx \leftarrow Rx - Ry$
LOAD Rx, Ry	R	0010	$Rx \leftarrow [Ry]$
STORE Rx, Ry	R	0011	$[Rx] \leftarrow Ry$
MOV Rx, Ry	R	0100	$Rx \leftarrow Ry$
NEG Rx	S	0101	$Rx \leftarrow \text{inverso aditivo de } Rx$
SKIP <sub>z</sub> Rx	S	0110	$PC \leftarrow PC + 1 \text{ si } Rx=0$
JMP Rx	S	0111	$PC \leftarrow Rx$

El formato de instrucción es el siguiente:

bits	7	6	5	4	3	2	1	0
	4 bits				2 bits		2 bits	
R	opCode				Rx		Ry	
	4 bits				2 bits		2 bits	
S	opCode				Ry		11	

a. Definir:

- ¿Cuál es el tamaño máximo de la memoria?
- El tamaño del PC
- Considerando que una instrucción inválida es una tira de bits que no codifica una instrucción que la máquina pueda ejecutar, determinar si los siguientes numerales codifican una instrucción válida: 0x82, 0x00, 0x44, 0x7C.

b. Se quiere ensamblar y cargar desde la posición de memoria 0x33 el siguiente código en una RECUP-LAST<sup>1L</sup>:

```

inicio: LOAD R0, R1
        NEG R1
        ADD R0, R1
        SKIPz R2
        STORE R0, R3
        JMP R2
  
```

- Definir en qué posición de memoria estará cada instrucción.
  - Indicar el contenido de la memoria luego de ensamblar y cargar el código anterior.
- c. Para cada parte de la planilla de seguimiento de la máquina ORGAL, justificar si es necesaria o no dicha celda. Indicar si es necesario agregar nuevas celdas o no para poder realizar el seguimiento.

- d. Modificar la planilla de seguimiento de la ORGA1 y realizar el seguimiento de la RECUP-LAST<sup>1L</sup> con la memoria presentada a continuación y mostrar que se detiene antes de ejecutar 8 instrucciones, sabiendo que el PC inicia en 0x50, R0 empieza en 0x2F, R1 en 0xDB, R2 en 0x50, R3 en 0x58, y la memoria toda en cero salvo las posiciones indicadas:

0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58	0x59
0x41	0x53	0x61	0x63	0x7B	0x7F	0x63	0x7F	0x82	0x33

### Ejercicio 2

- Armar un componente con compuertas lógicas que dado un número de 8 bits devuelva su inverso aditivo (comportamiento de la instrucción NEG).
- Realice el diagrama del *datapath* de una microarquitectura para la RECUP-LAST<sup>1L</sup> que soporte la ejecución de **TODAS** las instrucciones descriptas. Recuerde indicar el tamaño de cada registro, de los buses internos y externos, las señales de cada componente y justificar la utilización de cada componente escogido y cada decisión tomada.

Para realizar el *datapath* puede utilizar los siguientes componentes:

- un sumador completo.
- un único incrementador, con la operación sumar\_1.
- un comparador con cero.
- un único controlador de memoria.

Al incluirlos detallar cuidadosamente el tamaño de los registros y los nombres de las señales. Cualquier otro componente a utilizar deberá ser implementado e incluido en el examen.

- Escriba las microinstrucciones que debe ejecutar la máquina para realizar el *fetch* de una instrucción (no incluir etapas posteriores del ciclo de instrucción).
- Escriba el microprograma que realiza la parte de ejecución del ciclo de instrucción de las siguientes instrucciones:

- I. SUB R0, R3
- II. LOAD R1, R0
- III. SKIP\_Z R2
- IV. JMP R3

**Ejercicio 3** Los organizadores del mundial nos pidieron programar el controlador de las nuevas máquinas expendedoras de gaseosas MANN-AH-OSS. La misma utiliza una ORGA1i y tiene 3 dispositivos de E/S:

**EXP** (expendedor) que tiene un registro CAPACIDAD que, sólo si se acabaron las gaseosas, contiene 0xF177. Al modificarse este registro se informa con una interrupción

**ST** (sensor de temperatura) con un registro TEMP que indica la temperatura de la heladera en °C.

**ENF** (enfriador) con un registro de escritura: POTENCIA, que debe ser escrito periódicamente con el valor deseado. Los valores posibles son 0, 1 y 2.

El comportamiento esperado, para no congelar las bebidas, es enfriar las gaseosas sólo mientras la temperatura sea mayor o igual a 4°C. Para esto la potencia de acuerdo a la temperatura ( $t$ ) debe ser de la siguiente manera: 0 si  $t < 4^\circ$ ; 1 si  $4^\circ \leq t < 10^\circ$ ; 2 con  $t \geq 10^\circ$ .

Además cuando no haya más gaseosas se debe mantener el motor del ENF en 0 si  $t < 10$ , y 1 en caso contrario.

- Mapear los registros de los dispositivos de E/S e indicar para cada uno si es de lectura, escritura o lectura-escritura.
- Realizar el pseudo-código de la rutina principal del sistema y de la Rutina de Atención de Interrupciones (RAI) para aprovechar las capacidades provistas por los dispositivos.
- Realizar el código, en lenguaje ensamblador, de la rutina principal y de la RAI.